

// Fielder F at initial position P

agent fielder(F)(P) {

state

```

pos = P
speed = 10.0 // maximum speed 10 m/s
strength = 20.0 // can throw a ball at 20 m/s
range = 2.0 // can intercept ball within 2m radius

```

oracle

```

ball.pos
ball.velocity
pos
speed

```

derivate

```

exp_int_pt = intercept_pt(pos, speed, ball.pos, ball.velocity) // expected point of interception
interruptable = is_in_field(exp_int_pt)
time_to_intercept = intercept_time(ball.pos, exp_int_pt) / speed(ball.velocity)

```

role run() {

state

```

t0 = |Time|,
init_pos = |pos|
dest = |exp_int_pt|

```

derivate

```

//t
pos = init_pos + | speed * (dest - init_pos) dt
//t0
LIVE = interruptable &&  $\frac{d}{dt}(\text{time\_to\_interrupt}) \leq 1$  && !closeto(pos, ball.pos, range)
&& !closeto(pos, dest, range/2) && ball.deliverer != F

```

}

role intercept() {

handle

```
ball.owner
```

derivate

```
LIVE = closeto(pos, ball.pos, range) && ball.owner != F
```

protocol

```
closeto(@pos, @ball.pos, range) -> ball.owner = F
```

}

protocol

```

ball.owner == F
-> transfer(F)(velocity(pos, pavend, strength))

```

}

// throw the ball by the deliverer D at velocity V

process transfer(D)(V) {

state

```

t0 = |Time|
init_pos = | D.pos |
init_vel = | Velocity |
pos
velocity

```

handle

```
ball.owner = transfer(D)
```

derivate

```

//t
velocity = maxZ(init_vel - | g dt, 0)
//t0

```

```

//t
pos = init_pos + | velocity dt
//t0

```

```
LIVE = ball.owner == transfer(D)
```

}

fielder

run

intercept

// throw to pavend

transfer

```

object ball() {
state
    pos
    velocity
    owner // who is taking control over the ball
    deliverer // who has last touched the ball
derivate
    pos = owner.pos
    velocity = owner.velocity
}

```

ball

run

intercept

transfer